

#2 N-01007
(153452/00)

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

11000 U.S. PTO
09/862471
05/23/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2000年 5月24日

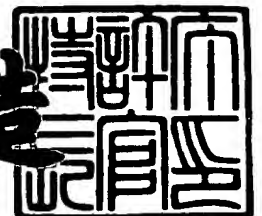
出 願 番 号
Application Number: 特願2000-153452

出 願 人
Applicant(s): 日本電気株式会社

2001年 2月16日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2001-3008574

【書類名】 特許願

【整理番号】 66206244

【提出日】 平成12年 5月24日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/38

【発明者】

 【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内

 【氏名】 川口 光治

【特許出願人】

 【識別番号】 000004237

 【氏名又は名称】 日本電気株式会社

【代理人】

 【識別番号】 100102864

 【弁理士】

 【氏名又は名称】 工藤 実

【選任した代理人】

 【識別番号】 100099553

 【弁理士】

 【氏名又は名称】 大村 雅生

【手数料の表示】

 【予納台帳番号】 053213

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9715177

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 命令バッファ及びバッファキュー制御

【特許請求の範囲】

【請求項 1】 順序が決定している命令からなる命令列と、
順序が決定しているエントリからなるバッファとを含み、
前記命令は、前記エントリに格納され、
順序が早い前記エントリに格納される前記命令は、順序が遅い前記エントリに
格納される前記命令より順序が早い
命令バッファ。

【請求項 2】 請求項 1 において、
前記エントリは、格納している前記命令が取り出し可能であるかどうかを示す
命令バッファ。

【請求項 3】 請求項 2 において、
前記取り出し可能である前記エントリのうち順序が最も早い前記エントリから
命令が取り出される
命令バッファ。

【請求項 4】 請求項 3 において、
前記命令を格納している前記エントリは、前記命令を格納していない前記エン
トリより順序が早い
命令バッファ。

【請求項 5】 請求項 4 において、
前記命令を格納する他のエントリからなる他のバッファを更に含み、
前記他のエントリに時間的に先に格納された命令は、前記他のエントリに時間
的に後に格納された命令より先に取り出される
命令バッファ。

【請求項 6】 順序が決定している複数の命令からなる第 1 命令群を生成する
こと、

前記第 1 命令群に属し実行可能である命令からなる第 2 命令群を生成すること

前記第 2 命令群に属し前記順序が最も早い命令を実行すること
とを含むバッファキュー制御。

【請求項 7】請求項 6 において、

前記第 1 命令群に包含される第 3 命令群を生成すること、

前記第 1 命令群に包含され前記第 3 命令群と互いに素である第 4 命令群を生成
することとを更に含み、

前記第 4 命令群に属し前記順序が最も早い命令が前記第 2 命令群に属さない
とき、前記第 4 命令群に属する全ての命令は実行されない
バッファキュー制御。

【請求項 8】請求項 7 において、

前記第 3 命令群または前記第 4 命令群に属する所定の 2 つの命令のうち 1 つの
命令が実行されるまで、前記 2 つの命令のうち他の 1 つの命令は実行可能ではな
い

バッファキュー制御。

【請求項 9】請求項 8 において、

前記第 3 命令群に属する命令は、前記第 4 命令群に属する命令と同時的に実行
される

バッファキュー制御。

【請求項 10】請求項 9 において、

前記第 3 命令群に属する命令は、演算系であり、

前記第 4 命令群に属する命令は、メモリアクセス系である

バッファキュー制御。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、命令バッファ及びバッファキュー制御に関し、特に、複数命令をよ
り高速に実行する命令バッファ及びバッファキュー制御に関する。

【0002】

【従来の技術】

命令をより高速に処理するために、各命令の実行を複数のステージに細分し任意の時点において1群の命令が同時に複数のステージにあることができるパイプライン処理が行われている。例えば、命令を読み出すステージ、命令を解読するステージ、命令を発行するステージ、命令の演算を実行するステージからなり、1つの命令が命令を解読するステージにあり、他の1つの命令が命令を発行するステージにある時点が存在する。

【0003】

複数の命令を発行する順序としてイン・オーダー発行とアウト・オブ・オーダー発行とがある。そのイン・オーダー発行は、命令をプログラムでの順序と同じ順序で発行することである。アウト・オブ・オーダー発行は、命令が実行可能になった順から命令を発行することである。このようなアウト・オブ・オーダー発行は、命令の発行効率を向上させ、演算器など使用効率を向上させ、全体の処理を速くする。

【0004】

このようなアウト・オブ・オーダー発行は、プログラムの順序と独立に命令が発行されるため、順序が早い命令が取り残される場合がある。このとき、取り残された命令と依存関係がある命令は発行できなくなり、命令バッファに全ての命令が登録され、全体の処理が遅くなってしまう。

【0005】

【発明が解決しようとする課題】

本発明の課題は、より高速に複数命令を発行できる命令バッファ及びバッファキュー制御を提供することにある。

【0006】

【課題を解決するための手段】

その課題を解決するための手段が、下記のように表現される。その表現中に現れる技術的事項には、括弧（ ）付きで、番号、記号等が添記されている。その番号、記号等は、本発明の実施の複数・形態又は複数の実施例のうちの少なくとも1つの実施の形態又は複数の実施例を構成する技術的事項、特に、その実施の形態又は実施例に対応する図面に表現されている技術的事項に付せられている参照

番号、参照記号等に一致している。このような参照番号、参照記号は、請求項記載の技術的事項と実施の形態又は実施例の技術的事項との対応・橋渡しを明確にしている。このような対応・橋渡しは、請求項記載の技術的事項が実施の形態又は実施例の技術的事項に限定されて解釈することを意味しない。

【0007】

本発明による命令バッファは、順序が決定している命令からなる命令列と、順序が決定しているエントリ（61～66）からなるバッファ（22）とを含み、命令は、エントリに格納され、順序が早いエントリに格納される命令は、順序が遅いエントリに格納される命令より順序が早い。エントリの順序と格納されている命令の順序とは一致している。

【0008】

エントリ（61～66）は、格納している命令が取り出し可能であるかどうかを示すことが好ましい。取り出し可能であるエントリに格納されている命令のうち順序の最も早いエントリに格納されている命令が取り出される。このため、順序の早い命令ほど、取り出される優先順位は高い。

【0009】

命令を格納しているエントリは、命令を格納していないエントリより順序が早い。命令を格納していないエントリがあり、そのエントリより順序が遅いエントリに格納されている命令は、命令を格納していないエントリに詰められる。

【0010】

命令を格納する他のエントリ（81～86）からなる他のバッファ（23）を更に含み、他のエントリ（81～86）に時間的に先に格納された命令は、他のエントリ（81～86）に時間的に後に格納された命令より先に取り出される。他のバッファ（23）に格納された命令のうち最も古い命令から順に取り出しが起こる。

【0011】

エントリ（61～66）に格納される命令は、演算系であり、他のエントリ（81～86）に格納される命令は、メモリアクセス系であることが好ましい。

【0012】

本発明によるバッファキュー制御は、順序が決定している複数の命令からなる第1命令群を生成すること、第1命令群に属し実行可能である命令からなる第2命令群を生成すること、第2命令群に属し順序が最も早い命令を実行することを含む。このように命令を実行することにより、順序が早い命令が実行されずに取り残されることはない。

【0013】

第1命令群に包含される第3命令群を生成すること、第1命令群に包含され第3命令群と互いに素である第4命令群を生成することとを更に含み、第4命令群に属し順序が最も早い命令が第2命令群に属さないとき、第4命令群に属する全ての命令は実行されない。第3命令群に属し且つ第4命令群に属する命令は存在しない。第4命令群に属する命令は、最も順序が早い命令から実行される。即ち、第4命令群に属する命令はイン・オーダー実行される。

【0014】

第3命令群または第4命令群に属する所定の2つの命令のうち1つの命令が実行されるまで、2つの命令のうち他の1つの命令は実行可能ではない。命令が実行可能であるかどうかは、所定の2つの命令の依存関係によって決定する。第3命令群に属する命令は、演算系である。第4命令群に属する命令は、メモリアクセス系である。演算系命令は、順序保証されていない命令である。メモリアクセス系命令は、順序保証してある命令である。第1命令群は、これら2種類に分割されることが好ましい。

【0015】

【発明の実施の形態】

図に一致対応して、本発明による命令バッファの実施の形態は、命令バッファが命令フェッチステージ、命令振り分けステージ、命令デコードステージ、命令登録ステージ、命令発行ステージ、命令実行ステージ及び命令完了ステージから構成されている。命令フェッチステージ1は、図1に示されるように、命令キャッシュメモリ11を備えている。命令キャッシュメモリ11は、命令を命令振り分けステージ2に供給する。

【0016】

命令振り分けステージ2は、フェッチレジスタ12を備えている。命令フェッチステージ1の命令キャッシュメモリ11は、フェッチレジスタ12に接続している。フェッチレジスタ12は、命令キャッシュメモリ11から供給された命令を蓄え、命令を演算系命令とメモリアクセス系命令とに振り分ける。

【0017】

命令デコードステージ3は、デコードレジスタ13とデコードレジスタ14とを備えている。命令振り分けステージ2のフェッチレジスタ12は、デコードレジスタ13に接続し、デコードレジスタ14に接続している。デコードレジスタ13には、演算系命令が登録される。演算系命令は、順序保証されていない命令である。演算系命令としては、四則演算、論理演算が例示される。デコードレジスタ14には、メモリアクセス系命令が登録される。メモリアクセス系命令は、順序保証してある命令であり、ロード命令、ストア命令が例示される。命令デコードステージ3は、更に、命令デコーダ15～18を備えている。デコードレジスタ13は、命令デコーダ15に接続し、命令デコーダ16に接続している。デコードレジスタ14は、命令デコーダ17に接続し、命令デコーダ18に接続している。命令デコーダ15、16は、デコードレジスタ13に登録された命令をデコードする。命令デコーダ17、18は、デコードレジスタ14に登録された命令をデコードする。命令デコーダ15～18は、命令登録報告31を生成する。命令登録報告31は、デコードされた命令が登録されることを示す。

【0018】

命令登録ステージ4は、リオーダバッファ（以下、「ROB」と記載される）21、演算系命令バッファ22及びメモリアクセス系命令バッファ23を備えている。命令デコードステージ3の命令デコーダ15～18は、ROB21に接続している。ROB21は、全ての命令を命令順に登録し、命令が完了した順に命令を解放する。ROB21は、命令順を保持し、命令間の依存関係の検出と命令の終了の確認とのために用いられる。演算系命令バッファ22には演算系命令が登録され、演算系命令バッファ22は演算系命令を発行する。メモリアクセス系命令バッファ23にはメモリアクセス系命令が登録され、メモリアクセス系命令バッファ23はメモリアクセス系命令を発行する。

【 0 0 1 9 】

命令発行ステージ5は、命令発行レジスタ24と命令発行レジスタ25とを備えている。演算系命令バッファ22は、命令発行レジスタ24に接続している。命令発行レジスタ24は、演算系命令バッファ22から発行された命令を蓄え、依存関係解放報告32を演算系命令バッファ22とメモリアクセス系命令バッファ23とに通知する。依存関係解放報告32は、どの命令が発行されたかを示す。メモリアクセス系バッファ23は、命令発行レジスタ25に接続している。命令発行レジスタ25はメモリアクセス系バッファ23から発行された命令を蓄え、依存関係解放報告32を演算系命令バッファ22とメモリアクセス系命令バッファ23とに通知する。

【 0 0 2 0 】

命令実行ステージ6は、命令実行レジスタ26と命令実行レジスタ27とを備えている。命令発行レジスタ24は命令実行レジスタ26に接続し、命令実行レジスタ26は演算系命令バッファ22に接続している。命令実行レジスタ26は、バッファ解放報告33を演算系命令バッファ22に通知する。バッファ解放報告33は、バッファを解放しても良いかどうかを示す。命令発行レジスタ25は命令実行レジスタ27に接続し、命令実行レジスタ27はメモリアクセス系命令バッファ23に接続している。命令実行レジスタ27は、バッファ解放報告34をメモリアクセス系命令バッファ23に通知する。バッファ解放報告34は、バッファを解放しても良いかどうかを示す。

【 0 0 2 1 】

命令完了ステージ7は、命令完了レジスタ28と命令完了レジスタ29とを備えている。命令実行レジスタ26は命令完了レジスタ28に接続し、命令完了レジスタ28はROB21に接続している。命令完了レジスタ28は、命令完了報告35をROB21に通知する。命令完了報告35は、命令が完了したかどうかを示す。命令実行レジスタ27は命令完了レジスタ29に接続し、命令完了レジスタ29はROB21に接続している。命令完了レジスタ29は、命令完了報告35をROB21に通知する。

【 0 0 2 2 】

図2は、ROB21の構成を示す。ROB21は、図2に示されるように、エントリ41～52を備えている。各エントリ41～52は、命令53、エントリ解放54及びエントリ有効55を登録している。エントリ解放54は、命令53を解放して良いかどうかを示す。エントリ有効55は、命令53が有効かどうかを示す。ROB21は、命令登録報告31を受けてエントリを有効にし、命令完了報告35を受けてエントリを解放する。

【0023】

図3は、演算系命令バッファ22の構成を示す。演算系命令バッファ22は、図3に示されるように、エントリ61～66とセクタ73～77とを備えている。命令デコーダ15、16は、エントリ66とセクタ73～77とに接続している。エントリ66はセクタ77に接続し、セクタ77はエントリ65に接続している。エントリ65はセクタ76に接続し、セクタ76はエントリ64に接続している。エントリ64はセクタ75に接続し、セクタ75はエントリ63に接続している。エントリ63はセクタ74に接続し、セクタ74はエントリ62に接続している。エントリ62はセクタ73に接続し、セクタ73はエントリ61に接続している。エントリ61～66は、命令発行レジスタ24に接続している。

【0024】

各エントリ61～66は、命令67、依存関係制御68、エントリ解放69及びエントリ有効71からなる。依存関係制御68は、そのエントリに属する命令67が他の命令と依存関係があるかどうかを示す。エントリ解放69は、そのエントリを解放して良いかどうかを示す。エントリ有効71は、そのエントリが有効かどうかを示す。

【0025】

セクタ73～77は、各エントリ61～66にそれぞれ命令を登録し更にバッファキュー制御を行う。即ち、セクタ73は、エントリ61が解放されたら、エントリ62の登録をエントリ61に移動する。セクタ74は、エントリ62が解放されたら、エントリ63の登録をエントリ62に移動する。セクタ75は、エントリ63が解放されたら、エントリ64の登録をエントリ63に移動

する。セクタ 7 6 は、エントリ 6 4 が解放されたら、エントリ 6 5 の登録をエントリ 6 4 に移動する。セクタ 7 7 は、エントリ 6 5 が解放されたら、エントリ 6 6 の登録をエントリ 6 5 に移動する。

【 0 0 2 6 】

新たな演算系命令は、有効でないエントリに登録される。バッファキュー制御と同時に演算系命令が登録されても良い。他の命令と依存関係がなく、添付番号が最も小さいエントリに属する演算系命令が発行される。演算系命令バッファ 2 2 は、更に、発行ポインタ 7 2 を備えている。発行ポインタ 7 2 は、演算系命令の発行を制御する。バッファキュー制御と同時に演算系命令が発行されても良い。

【 0 0 2 7 】

図 4 は、メモリアクセス系バッファ 2 3 の構成を示す。メモリアクセス系命令バッファ 2 3 は、図 4 に示されるように、エントリ 8 1 ～ 8 6 を備えている。命令デコーダ 1 7, 1 8 は、エントリ 8 1 ～ 8 6 に接続している。エントリ 8 1 ～ 8 6 は、命令発行レジスタ 2 5 に接続している。各エントリ 8 1 ～ 8 6 は、命令 8 7、依存関係制御 8 8、エントリ解放 9 9 及びエントリ有効 9 1 を登録している。依存関係制御 8 8 は、そのエントリに属する命令が他の命令と依存関係があるかどうかを示す。エントリ解放 8 9 は、そのエントリを解放して良いかどうかを示す。エントリ有効 9 1 は、そのエントリが有効かどうかを示す。

【 0 0 2 8 】

メモリアクセス系バッファ 2 2 は、更に、発行ポインタ 9 2 と先頭ポインタ 9 3 とを備えている。発行ポインタ 9 2 は、命令の発行を制御する。他の命令と依存関係がなく、最初に登録された命令が発行される。即ち、イン・オーダ発行される。先頭ポインタ 9 3 は、どのエントリの命令までが発行されたかを示す。

【 0 0 2 9 】

図 5 は、本発明によるバッファキュー制御方法の実施の形態を示す。バッファキュー制御は、3 つのタイミングから形成される。タイミング T 1 は、命令が選択され命令発行レジスタ 2 4 にその命令が発行された状態である。タイミング T 2 は、命令が発行されてから数クロック進んだ状態であり、エントリが解放され

た状態である。タイミングT3は、キュー制御された状態である。演算系命令ALU-1~6は、エントリ61~66に登録される命令である。

【0030】

タイミングT1では、図5(a)に示されるように、エントリ61には、演算系命令ALU-1が登録されている。エントリ62には、演算系命令ALU-2が登録されている。エントリ63には、演算系命令ALU-3が登録されている。エントリ64には、演算系命令ALU-4が登録されている。エントリ65には、演算系命令ALU-5が登録されている。エントリ66には、演算系命令ALU-6が登録されている。このとき、演算系命令ALU-3は選択され命令発行レジスタ24に発行されている。

【0031】

タイミングT2では、図5(b)に示されるように、エントリ63は、演算系命令ALU-3が解放され空になっている。タイミングT3では、図5(c)に示されるように、演算系命令ALU-4はエントリ63に登録され、演算系命令ALU-5はエントリ64に登録され、演算系命令ALU-6はエントリ65に登録される。エントリ66は、空になっている。

【0032】

図6は、バッファキュー制御と同時的に演算系命令が登録される動作を示す。タイミングT1では、図6(a)に示されるように、エントリ61には、演算系命令ALU-1が登録されている。エントリ62には、演算系命令ALU-2が登録されている。エントリ63には、演算系命令ALU-3が登録されている。エントリ64~66は、空になっている。デコードレジスタ13は、空になっている。このとき、演算系命令ALU-1は選択され命令発行レジスタ24に発行されている。

【0033】

タイミングT2では、図6(b)に示されるように、エントリ61は、演算系命令ALU-1が解放され空になっている。デコードレジスタ13には、演算系命令ALU-4と演算系命令ALU-5とが登録される。

【0034】

タイミングT3では、図6(c)に示されるように、演算系命令ALU-2はエントリ61に登録され、演算系命令ALU-3はエントリ62に登録され、演算系命令ALU-4はエントリ63に登録され、演算系命令ALU-5はエントリ64に登録される。このように命令の登録は、バッファキュー制御と同時的に実行が可能である。

【0035】

図7は、バッファキュー制御と同時的に演算系命令が発行される動作を示す。タイミングT1では、図7(a)に示されるように、エントリ61には、演算系命令ALU-1が登録されている。エントリ62には、演算系命令ALU-2が登録されている。エントリ63には、演算系命令ALU-3が登録されている。エントリ64～66は、空になっている。演算系命令ALU-1は、選択され命令発行レジスタ24に発行されている。

【0036】

タイミングT2では、図7(b)に示されるように、エントリ61は、演算系命令ALU-1が解放され空になっている。タイミングT3では、図7(c)に示されるように、演算系命令ALU-2はエントリ61に登録され、演算系命令ALU-3はエントリ62に登録される。演算系命令ALU-2は、選択され命令発行レジスタ24に発行されている。このように命令の発行は、バッファキュー制御と同時的に実行できる。

【0037】

図8は、バッファキュー制御しないときの動作を示す。タイミングT1では、図8(a)に示されるように、エントリ61には、演算系命令ALU-1が登録されている。エントリ62には、演算系命令ALU-2が登録されている。エントリ63には、演算系命令ALU-3が登録されている。エントリ64には、演算系命令ALU-4が登録されている。エントリ65、66は、空になっている。演算系命令ALU-2は、選択され命令発行レジスタ24に発行されている。

【0038】

タイミングT2では、図8(b)に示されるように、エントリ62は、演算系命令ALU-2が解放され空になっている。デコードレジスタ13には、演算系

命令ALU-5と演算系命令ALU-6とが登録される。タイミングT3では、図8(c)に示されるように、演算系命令ALU-5はエントリ62に登録され、演算系命令ALU-6はエントリ65に登録される。デコードレジスタ13は空になる。

【0039】

このようにバッファキュー制御しないとき、演算系命令ALU-5は演算系命令ALU-3, ALU-4より後に登録されたにもかかわらず、優先順位は上位である。命令の順序は保持されず、バラバラである。

【0040】

図9は、バッファキュー制御したときの演算系命令バッファ22とメモリアクセス系バッファ23とを示す。依存関係制御68が1のとき、他の命令が発行された後にその命令が発行されなければならないことを示し、依存関係制御68が0のとき、発行されて良いことを示す。エントリ解放69が1のとき、そのエントリを解放しても良いことを示し、エントリ解放69が0のとき、そのエントリを解放してはいけないことを示す。エントリ有効71が1のとき、そのエントリに命令が登録されていることを示し、エントリ有効71が0のとき、そのエントリに登録可能であることを示す。

【0041】

演算系命令ALU-7は、エントリ61に登録されている。演算系命令ALU-5は、エントリ62に登録されている。演算系命令ALU-3は、エントリ63に登録されている。演算系命令ALU-4は、エントリ64に登録されている。演算系命令ALU-6は、エントリ65に登録されている。演算系命令ALU-8は、エントリ66に登録されている。

【0042】

依存関係制御88が1のとき、他の命令が発行された後にその命令が発行されなければならないことを示し、依存関係制御88が0のとき、発行されて良いことを示す。エントリ解放89が1のとき、そのエントリを解放しても良いことを示し、エントリ解放89が0のとき、そのエントリを解放してはいけないことを示す。エントリ有効91が1のとき、そのエントリに命令が登録されていること

を示し、エントリ有効 9 1 が 0 のとき、そのエントリに命令が登録可能であることを示す。

【 0 0 4 3 】

メモリアクセス系命令 MEM-1 は、エントリ 8 1 に登録されている。メモリアクセス系命令 MEM-2 は、エントリ 8 2 に登録されている。メモリアクセス系命令 MEM-3 は、エントリ 8 3 に登録されている。エントリ 8 4 ~ 8 6 は空になっている。

【 0 0 4 4 】

メモリアクセス系命令 MEM-1 は、依存関係を有し、演算系命令 ALU-3 が発行された後発行される。メモリアクセス系命令 MEM-2 は、依存関係を有し、演算系命令 ALU-4 が発行された後発行される。メモリアクセス系命令 MEM-3 は、依存関係を有し、演算系命令 ALU-5 が発行された後発行される。

【 0 0 4 5 】

図 1 0 のタイムチャートは、図 9 に示されるように命令が登録されたときのメモリアクセス系命令 MEM-1 ~ 3 と演算系命令 ALU-3 ~ 8 との状態を示す。タイムチャート中の R は命令登録ステージ 4 を示し、A I は命令発行ステージ 5 を示し、A X は命令実行ステージ 6 を示し、A W は命令完了ステージ 7 を示している。タイムチャート中の R は命令登録ステージ 4 を示し、E I は命令発行ステージ 5 を示し、E X は命令実行ステージ 6 を示し、E W は命令完了ステージ 7 を示している。

【 0 0 4 6 】

メモリアクセス系命令 MEM-1 ~ 3 は、依存関係があるので依存関係解放報告 3 2 が通知されないと発行されず、且つ、エントリの番号の最も若い順に発行される。演算系命令 ALU-3 ~ 8 は、依存関係がないためエントリの添付番号の若い順に発行される。

【 0 0 4 7 】

タイミング T 2 では、演算系命令 ALU-7 が発行される。タイミング T 3 では、演算系命令 ALU-7 が実行される。演算系命令 ALU-5 が発行され、依

存関係解放報告32が通知される。タイミングT4では、演算系命令ALU-7は完了する。演算系命令ALU-5は実行される。演算系命令ALU-3は発行される。依存関係解放通知32が通知される。依存関係解放報告32を受けてメモリアクセス系命令MEM-3の依存関係が解放される。

【0048】

タイミングT5では、演算系命令ALU-5は完了する。演算系命令ALU-3は実行される。演算系命令ALU-4は発行される。依存関係解放通知32が通知される。依存関係解放報告32を受けてメモリアクセス系命令MEM-1の依存関係が解放される。

【0049】

タイミングT6では、演算系命令ALU-3は完了する。演算系命令ALU-4は実行される。演算系命令ALU-6は発行される。メモリアクセス系命令MEM-1は発行される。依存関係解放報告32を受けてメモリアクセス系命令MEM-2の依存関係が解放される。

【0050】

タイミングT7では、演算系命令ALU-4は完了する。演算系命令ALU-6は実行される。演算系命令ALU-8は発行される。メモリアクセス系命令MEM-1は実行される。メモリアクセス系命令MEM-2は発行される。タイミングT8では、演算系命令ALU-6は完了する。演算系命令ALU-8は実行される。メモリアクセス系命令MEM-1は完了する。メモリアクセス系命令MEM-2は実行される。メモリアクセス系命令MEM-3は発行される。

【0051】

タイミングT9では、演算系命令ALU-8は完了する。メモリアクセス系命令MEM-2は完了する。メモリアクセス系命令MEM-3は実行される。タイミングT10では、メモリアクセス系命令MEM-3は完了する。このように全ての命令が終了するまでには10タイミングを要する。

【0052】

図11は、図10のタイムチャートにおけるタイミングT8（図10中の矢印の部分）のときのROB21の状態を示す。エントリ41には、演算系命令AL

U-1が登録されている。エントリ42には、演算系命令ALU-2が登録されている。エントリ43には、演算系命令ALU-3が登録されている。エントリ44には、メモリアクセス系命令MEM-1が登録されている。エントリ45には、演算系命令ALU-4が登録されている。エントリ46には、メモリアクセス系命令MEM-2が登録されている。エントリ47には、演算系命令ALU-5が登録されている。エントリ48には、メモリアクセス系命令MEM-3が登録されている。エントリ49には、演算系命令ALU-6が登録されている。エントリ50には、演算系命令ALU-7が登録されている。エントリ51には、演算系命令ALU-8が登録されている。エントリ52は、空になっている。

【0053】

ROB21のエントリ41～52は、先に登録された命令が解放されない限り解放されない。エントリ47, 49, 50は、解放可能であるにもかかわらず、エントリ46が解放されていないため先頭ポインタ56は、エントリ46を指している。

【0054】

図12は、バッファキュー制御したときの演算系命令バッファ22とメモリアクセス系バッファ23とを示す。演算系命令ALU-3は、エントリ61に登録されている。演算系命令ALU-4は、エントリ62に登録されている。演算系命令ALU-5は、エントリ63に登録されている。演算系命令ALU-6は、エントリ64に登録されている。演算系命令ALU-7は、エントリ65に登録されている。演算系命令ALU-8は、エントリ66に登録されている。メモリアクセス系命令MEM-1は、エントリ81に登録されている。メモリアクセス系命令MEM-2は、エントリ82に登録されている。メモリアクセス系命令MEM-3は、エントリ83に登録されている。エントリ84～86は空になっている。

【0055】

メモリアクセス系命令MEM-1は、依存関係を有し、演算系命令ALU-3が発行された後発行される。メモリアクセス系命令MEM-2は、依存関係を有し、演算系命令ALU-4が発行された後発行される。メモリアクセス系命令M

EM-3は、依存関係を有し、演算系命令ALU-5が発行された後発行される。

【0056】

図13のタイムチャートは、図12に示されるように命令が登録されたときのメモリアクセス系命令MEM-1～3と演算系命令ALU-3～8との状態を示す。メモリアクセス系命令MEM-1～3は、依存関係があるので依存関係解放報告32が通知されないと発行されず、且つ、エントリの添付番号の最も若い順に発行される。演算系命令ALU-3～8は、依存関係がないためエントリの番号の若い順に発行される。

【0057】

タイミングT2では、演算系命令ALU-3が発行される。依存関係解放報告32が通知される。タイミングT3では、演算系命令ALU-3が実行される。演算系命令ALU-4が発行され、依存関係解放報告32が通知される。依存関係解放報告32を受けてメモリアクセス系命令MEM-1の依存関係が解放される。

【0058】

タイミングT4では、演算系命令ALU-3は完了する。演算系命令ALU-4は実行される。演算系命令ALU-5は発行される。依存関係解放通知32が通知される。メモリアクセス系命令MEM-1は発行される。依存関係解放報告32を受けてメモリアクセス系命令MEM-2の依存関係が解放される。

【0059】

タイミングT5では、演算系命令ALU-4は完了する。演算系命令ALU-5は実行される。演算系命令ALU-6は発行される。メモリアクセス系命令MEM-1は実行される。メモリアクセス系命令MEM-2は発行される。依存関係解放報告32を受けてメモリアクセス系命令MEM-3の依存関係が解放される。

【0060】

タイミングT6では、演算系命令ALU-5は完了する。演算系命令ALU-6は実行される。演算系命令ALU-7は発行される。メモリアクセス系命令M

EM-1は完了する。メモリアクセス系命令MEM-2は実行される。メモリアクセス系命令MEM-3は発行される。

【0061】

タイミングT7では、演算系命令ALU-6は完了する。演算系命令ALU-7は実行される。演算系命令ALU-8は発行される。メモリアクセス系命令MEM-2は完了する。メモリアクセス系命令MEM-3は実行される。

【0062】

タイミングT8では、演算系命令ALU-7は完了する。演算系命令ALU-8は実行される。メモリアクセス系命令MEM-3は完了する。タイミングT9では、演算系命令ALU-8が完了する。このように全ての命令が終了するまでには9タイミングを要する。

【0063】

図14は、図12のタイムチャートにおけるタイミングT8（図12中の矢印の部分）のときのROB21の状態を示す。エントリ41には、演算系命令ALU-1が登録されている。エントリ42には、演算系命令ALU-2が登録されている。エントリ43には、演算系命令ALU-3が登録されている。エントリ44には、メモリアクセス系命令MEM-1が登録されている。エントリ45には、演算系命令ALU-4が登録されている。エントリ46には、メモリアクセス系命令MEM-2が登録されている。エントリ47には、演算系命令ALU-5が登録されている。エントリ48には、メモリアクセス系命令MEM-3が登録されている。エントリ49には、演算系命令ALU-6が登録されている。エントリ50には、演算系命令ALU-7が登録されている。エントリ51には、演算系命令ALU-8が登録されている。エントリ52は、空になっている。

【0064】

このとき、エントリ41～50は解放可能である。先のバッファキュー制御しないときより、ROB21の資源を効率よく使用することができる。

【0065】

本発明によるバッファキュー制御によれば、古い命令の優先順位を高く設定することができる。このため、イン・オーダ発行されるメモリアクセス系命令に対

して速く依存関係が解消される。

【 0 0 6 6 】

【発明の効果】

本発明による命令バッファ及びバッファキュー制御は、より高速に複数命令を発行できる。

【図面の簡単な説明】

【図 1】

図 1 は、本発明による命令バッファの実施の形態を示すブロック図である。

【図 2】

図 2 は、R O B を示すブロック図である。

【図 3】

図 3 は、演算系命令バッファを示すブロック図である。

【図 4】

図 4 は、メモリアクセス系命令バッファを示すブロック図である。

【図 5】

図 5 は、演算系命令バッファの動作を示すタイムチャートである。

【図 6】

図 6 は、演算系命令バッファの動作を示すタイムチャートである。

【図 7】

図 7 は、演算系命令バッファの動作を示すタイムチャートである。

【図 8】

図 8 は、演算系命令バッファの動作を示すタイムチャートである。

【図 9】

図 9 は、制御しない場合の演算系命令バッファとメモリアクセス系命令バッファの状態を示すブロック図である。

【図 1 0】

図 1 0 は、制御しない場合の各命令の状態を示すタイムチャートである。

【図 1 1】

図 1 1 は、制御しない場合の R O B を示すブロック図である。

【図12】

図12は、制御した場合の演算系命令バッファとメモリアクセス系命令バッファを示す図である。

【図13】

図13は、制御した場合の各命令の状態を示すタイムチャートである。

【図14】

図14は、制御した場合のROBを示すブロック図である。

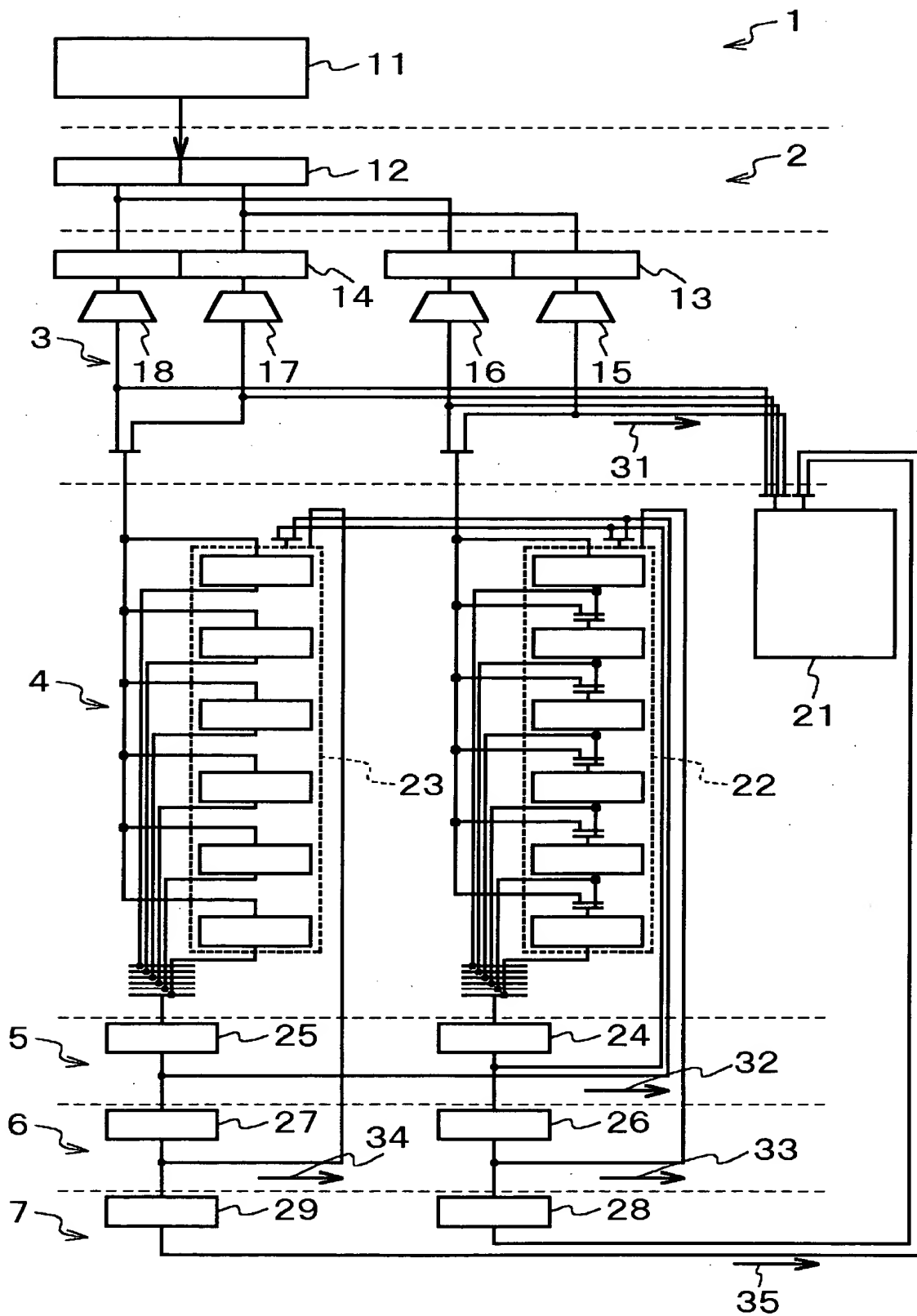
【符号の説明】

- 1…命令フェッチステージ
- 2…命令振り分けステージ
- 3…命令デコードステージ
- 4…命令登録ステージ
- 5…命令発行ステージ
- 6…命令実行ステージ
- 7…命令完了ステージ
- 11…命令キャッシュメモリ
- 12…フェッチレジスタ
- 13, 14…デコードレジスタ
- 15, 16, 17, 18…命令デコーダ
- 21…ROB
- 22…演算系命令バッファ
- 23…メモリアクセス系命令バッファ
- 24, 25…命令発行レジスタ
- 26, 27…命令実行レジスタ
- 28, 29…命令完了レジスタ
- 31…命令登録報告
- 32…依存関係解放報告
- 33, 34…バッファ解放報告
- 35…命令完了報告

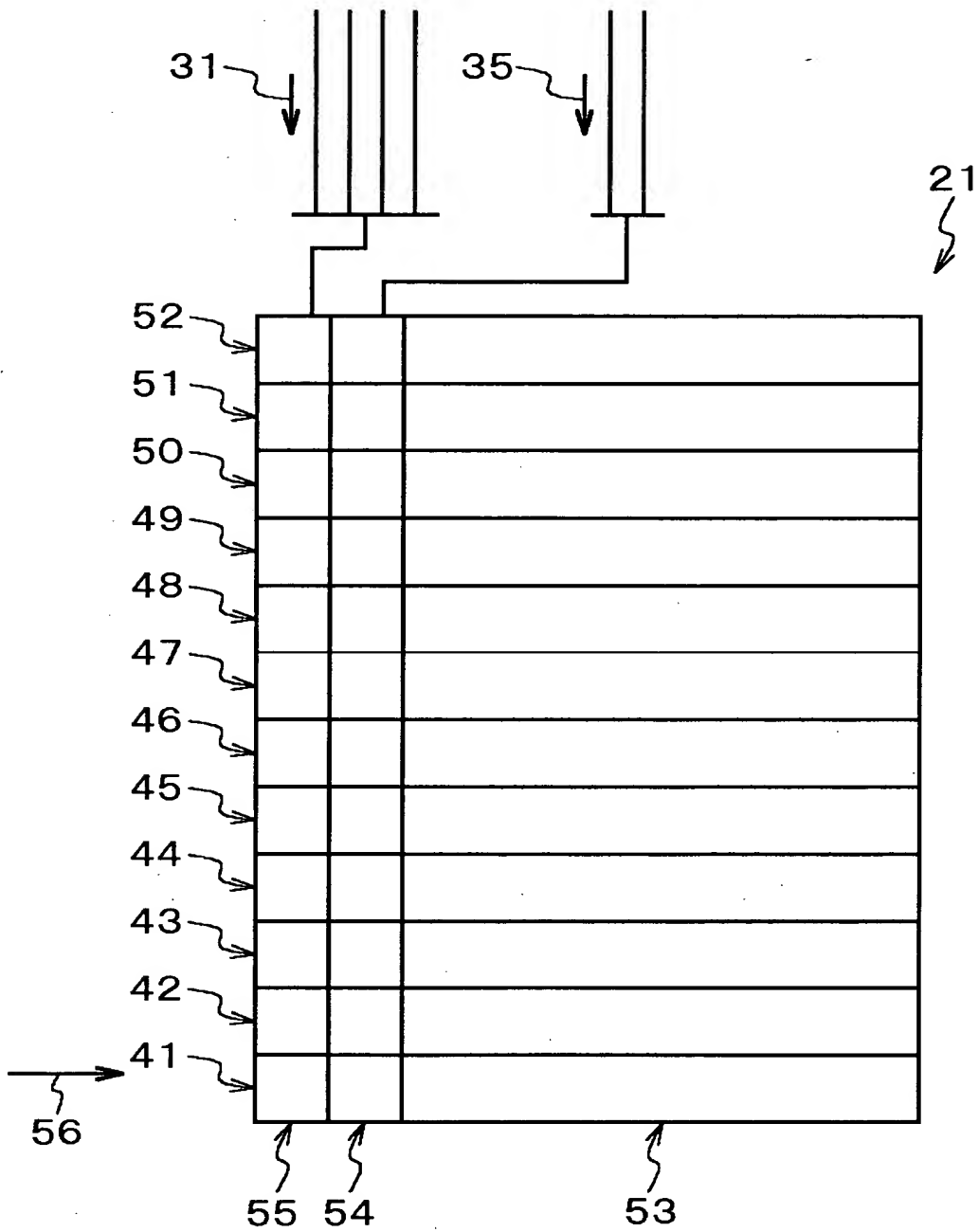
特 2 0 0 0 - 1 5 3 4 5 2

【書類名】 図面

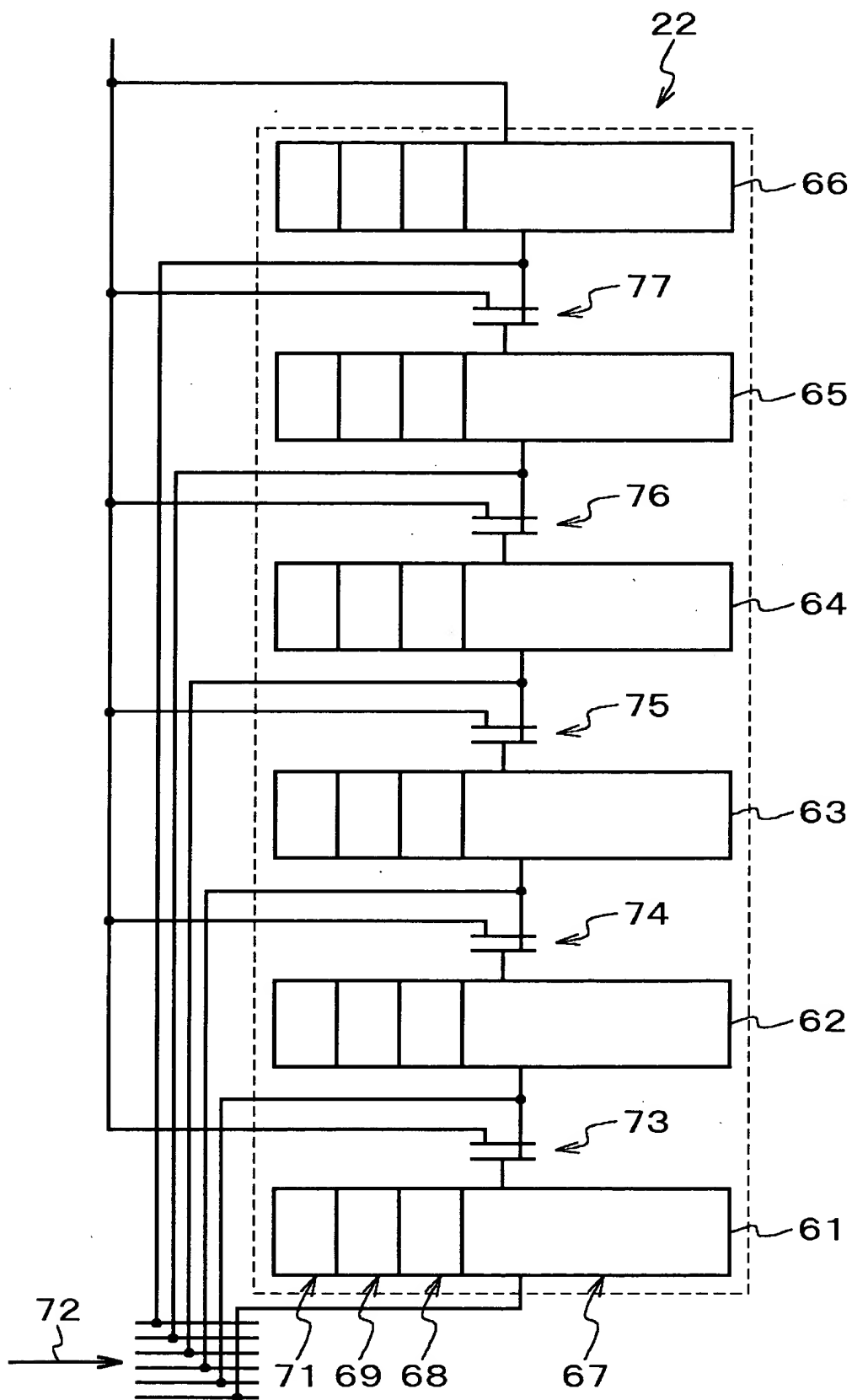
【図1】



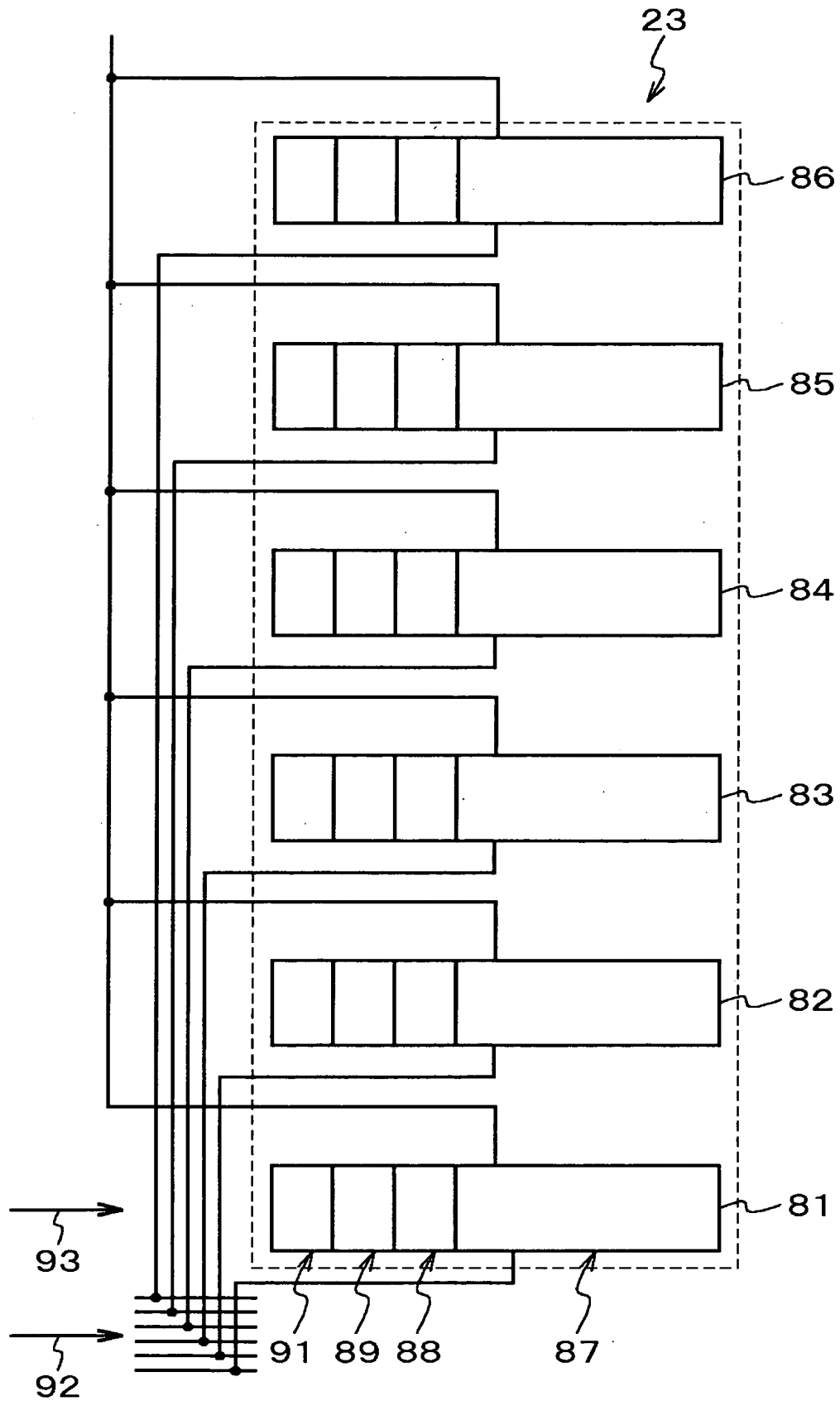
【図 2】



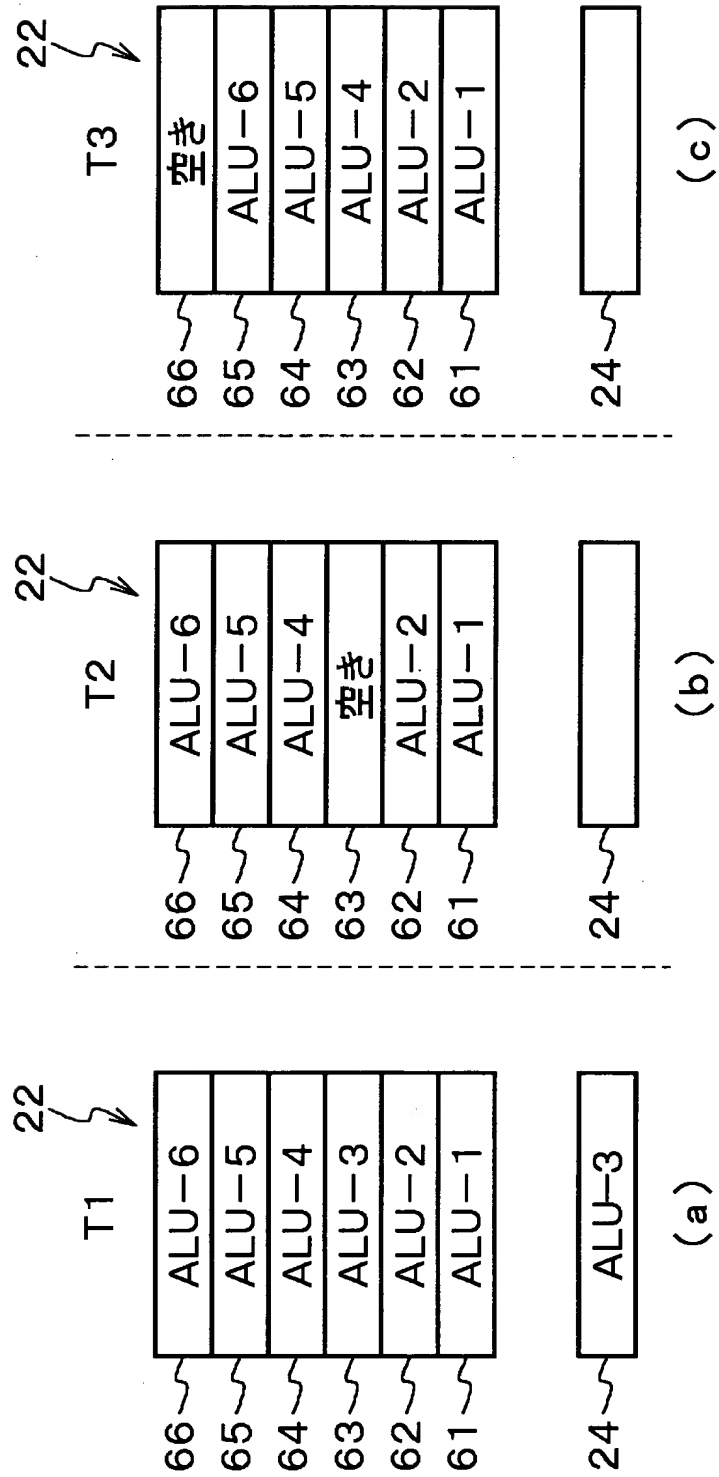
【図 3】



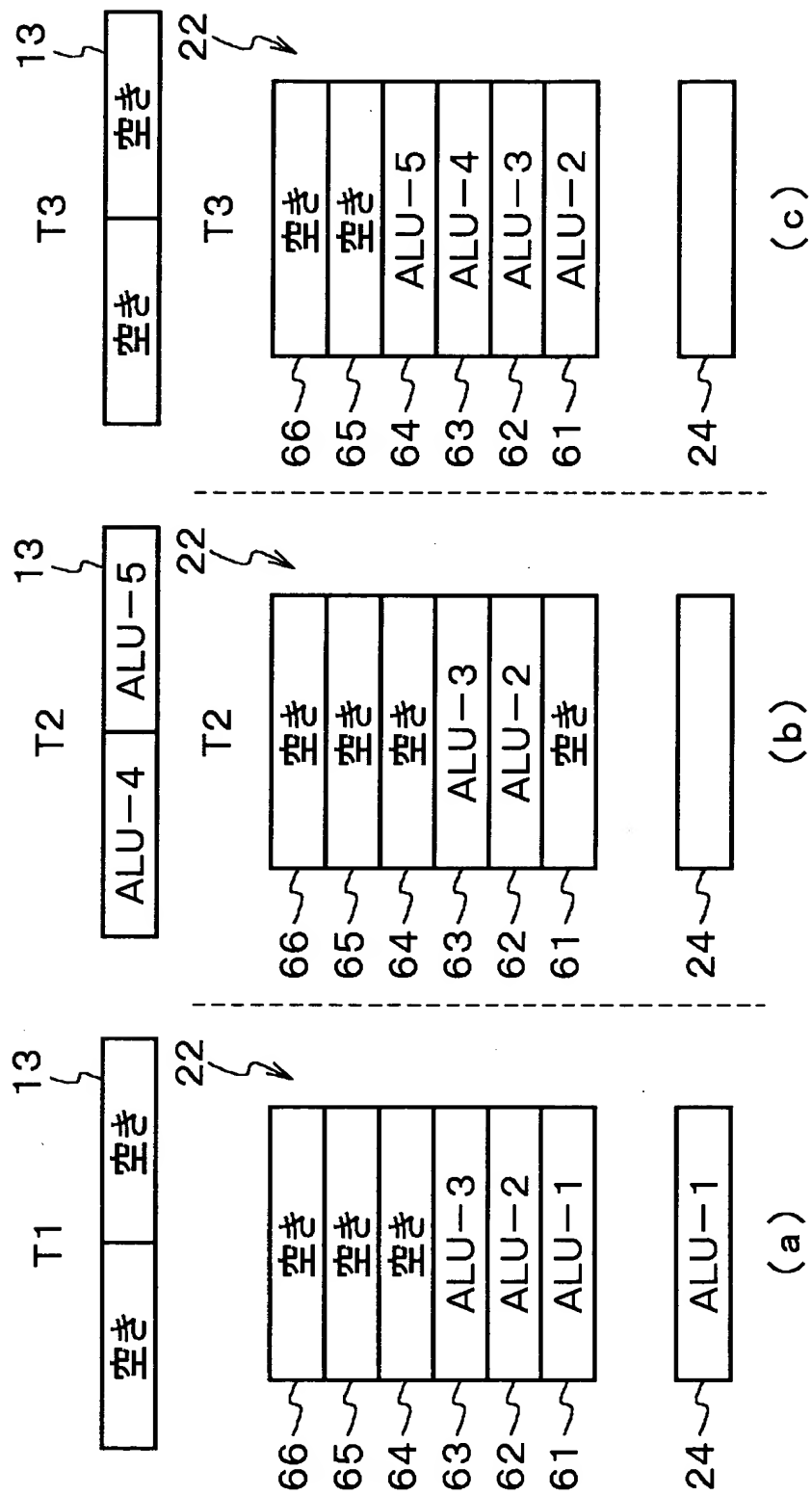
【図 4】



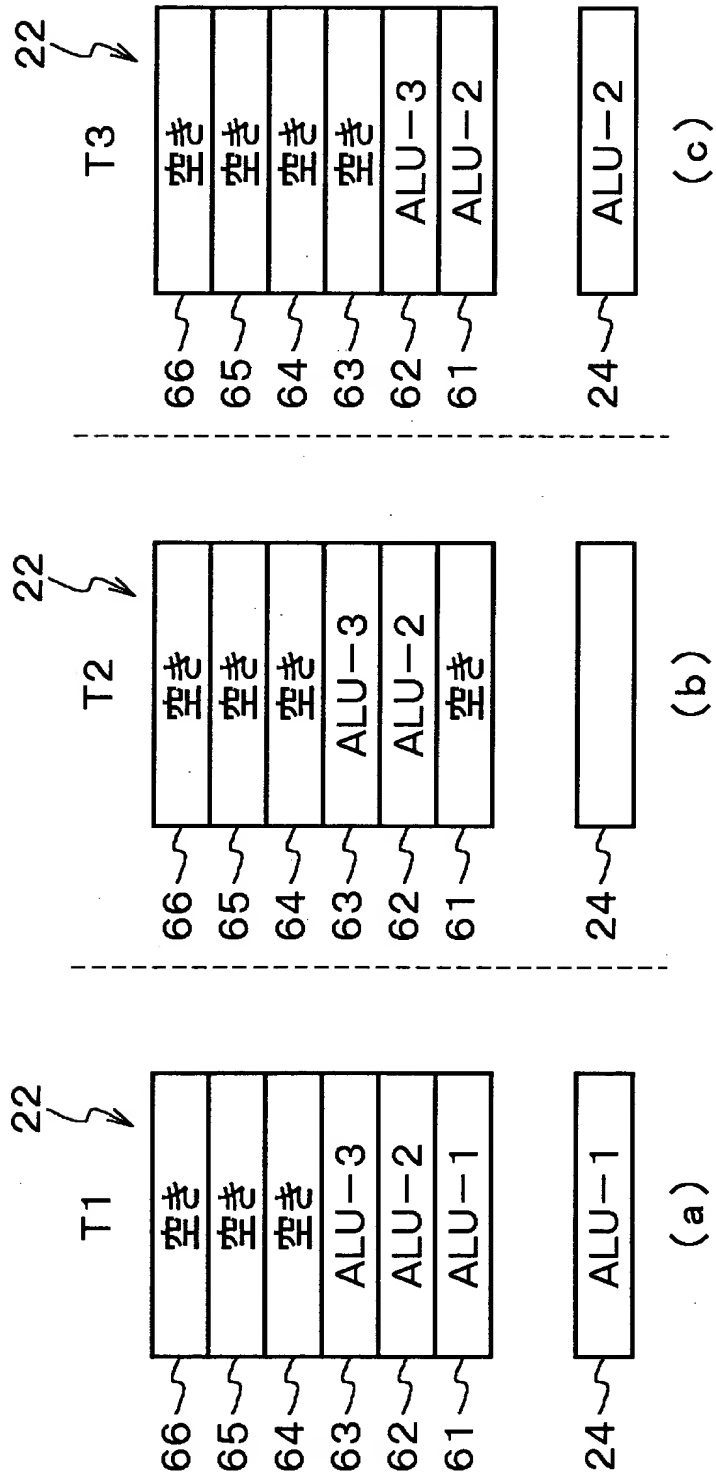
【図 5】



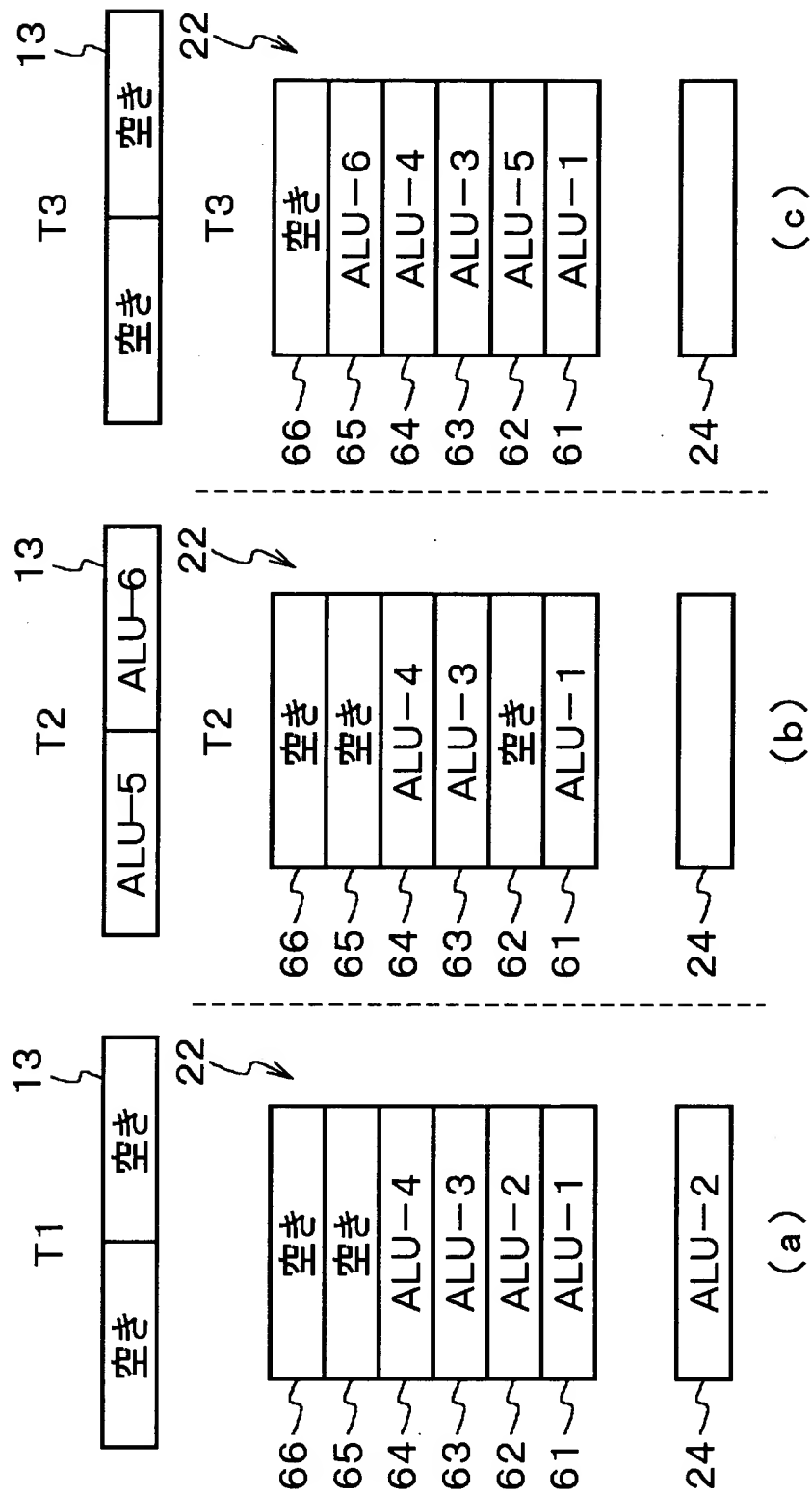
【図 6】



【図 7】



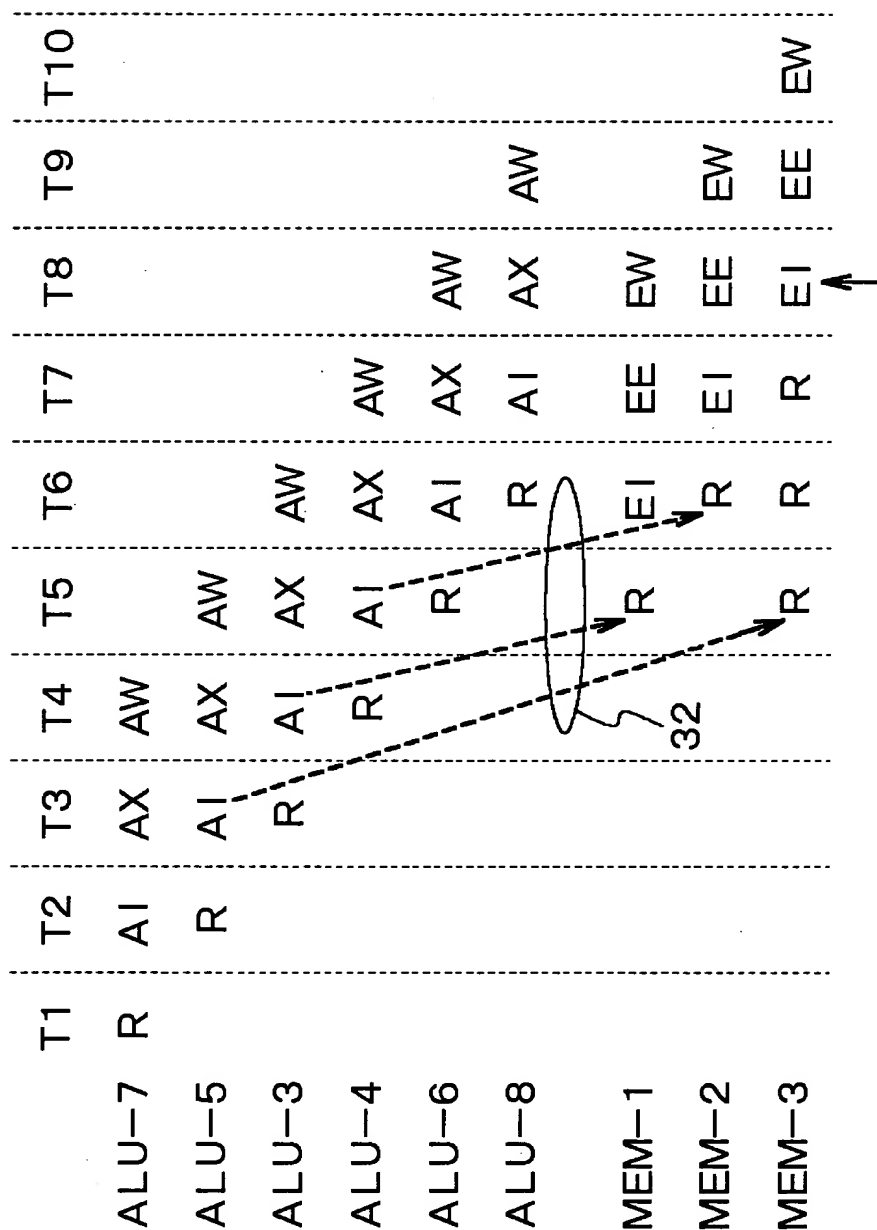
【図 8】



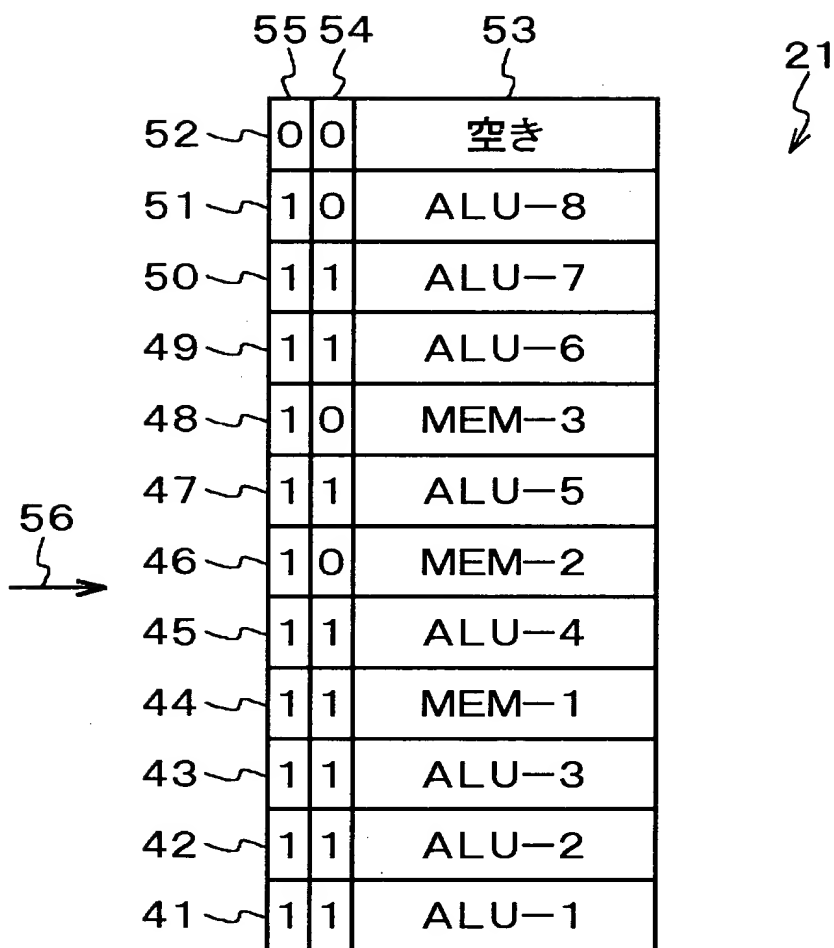
【図 9】

	91	89	88	87	23	71	69	68	67	22
86	0	0	0	空き		66	1	0	0	ALU-8
85	0	0	0	空き		65	1	0	0	ALU-6
84	0	0	0	空き		64	1	0	0	ALU-4
83	1	0	1	MEM-3		63	1	0	0	ALU-3
82	1	0	1	MEM-2		62	1	0	0	ALU-5
81	1	0	1	MEM-1		61	1	0	0	ALU-7

【図 10】



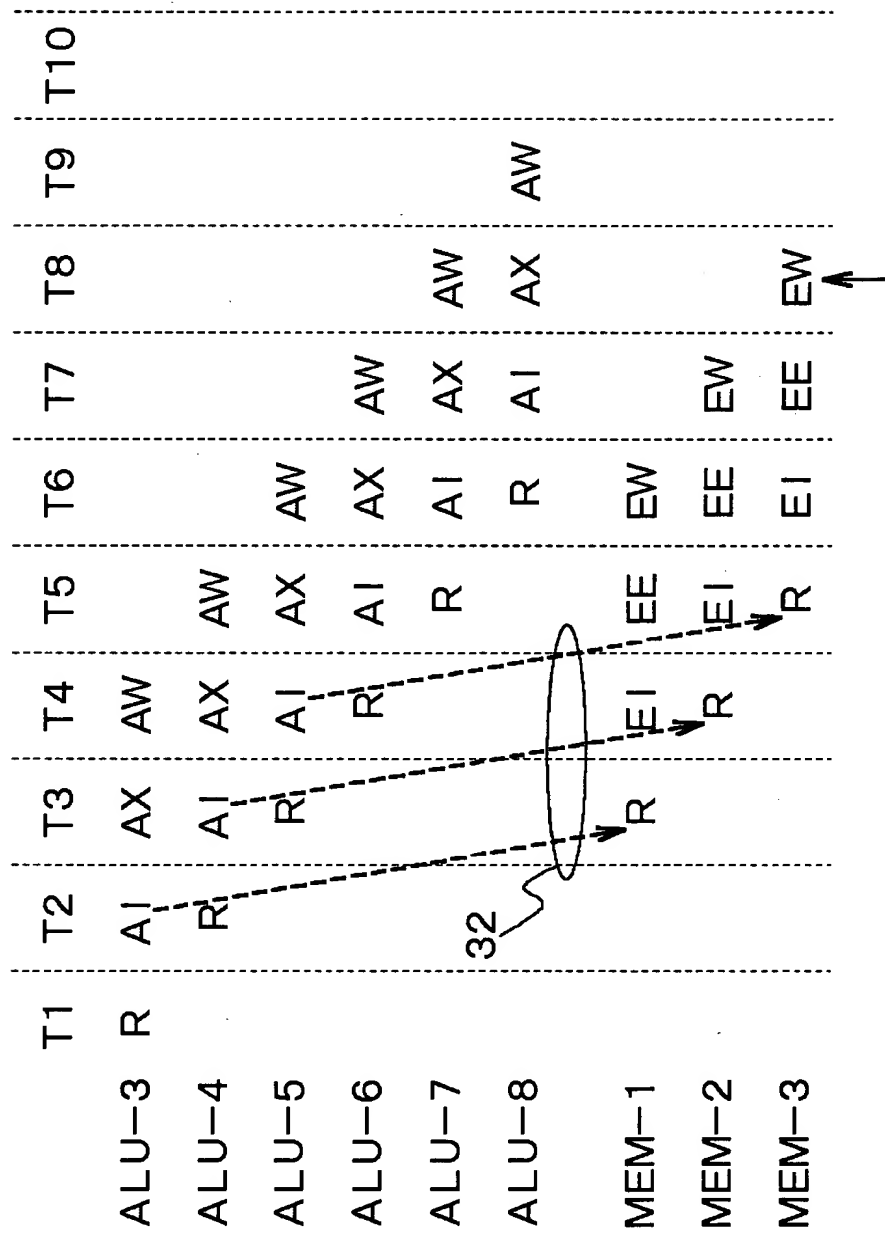
【図 1 1】



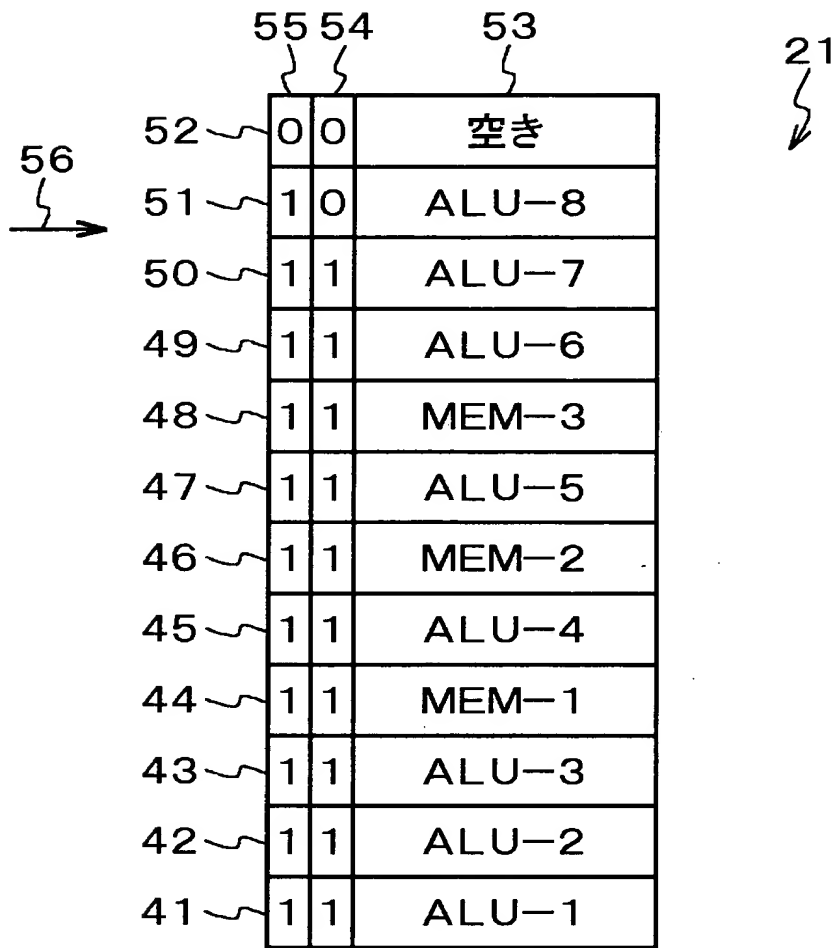
【図 1 2】

91	89	88	87	23	71	69	68	67	22
86	0	0	0	空き	66	1	0	0	ALU-8
85	0	0	0	空き	65	1	0	0	ALU-7
84	0	0	0	空き	64	1	0	0	ALU-6
83	1	0	1	MEM-3	63	1	0	0	ALU-5
82	1	0	1	MEM-2	62	1	0	0	ALU-4
81	1	0	1	MEM-1	61	1	0	0	ALU-3

【図 13】



【図 1 4】



【書類名】 要約書

【要約】

【課題】 より高速に複数の命令を発行する。

【解決手段】 順序が決定している命令からなる命令列と、順序が決定しているエントリ 6 1 ～ 6 6 からなるバッファ 2 2 とを含み、命令は、エントリに格納され、エントリの順序と格納されている命令の順序とは一致している。エントリ 6 1 ～ 6 6 は、格納している命令が取り出し可能であるかどうかを示し、命令が取り出し可能であるエントリのうち順序の最も早いエントリに格納されている命令が取り出される。命令を格納していないエントリがあり、そのエントリより順序が遅いエントリに格納されている命令は、命令を格納していないエントリに詰められる。命令を格納する他のエントリ 8 1 ～ 8 6 からなる他のバッファ 2 3 を更に含み、他のエントリ 8 1 ～ 8 6 に時間的に先に格納された命令は、後に格納された命令より先に取り出される。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2 0 0 0 - 1 5 3 4 5 2
受付番号	5 0 0 0 0 6 4 1 0 4 9
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 2 年 5 月 2 5 日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000004237
【住所又は居所】	東京都港区芝五丁目 7 番 1 号
【氏名又は名称】	日本電気株式会社

【代理人】

申請人

【識別番号】	100102864
【住所又は居所】	東京都品川区南大井 6 丁目 2 4 番 1 0 号 カドヤ 第 1 0 ビル 6 階 工藤国際特許事務所
【氏名又は名称】	工藤 実

【選任した代理人】

【識別番号】	100099553
【住所又は居所】	東京都品川区南大井 6 丁目 2 4 番 1 0 号 カドヤ 第 1 0 ビル 6 階 工藤国際特許事務所
【氏名又は名称】	大村 雅生

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社